

ABSTRACT

This project outlines the development of a web-based Restaurant Order System by Sensation Software Solutions Pvt Ltd, to revolutionize the dining experience by providing a comprehensive and user-friendly platform for customers, restaurants, and super admins. Built on modern technologies such as HTML, CSS, JavaScript, React, MongoDB, Express, and Node.js, the project offers seamless integration of functionalities tailored to each user.

Customer Functionality:

Customers benefit from a seamless ordering process, allowing them to browse menus, search for specific dishes, customize orders with options and add-ons, and place orders with ease. User accounts enable personalized recommendations, order history tracking, and participation in loyalty programs. Additionally, customers can leave reviews and ratings to share experiences and guide other customers, enhancing the overall dining experience.

Restaurant Functionality:

Restaurants have access to robust management tools, including the ability to add, edit, and manage restaurant information such as descriptions, operating hours, and contact details. Menu management features enable restaurants to create and update menus with detailed descriptions, offers, and pricing. Efficient order processing allows restaurants to receive and manage orders, track order status, confirm orders with customers, and communicate any updates seamlessly. The system also offers various print receipt options tailored for chefs, customers, and restaurant records. Furthermore, restaurants can leverage billing and analytics functionalities to generate and manage invoices, track sales performance, analyze customer behavior and order trends, and gain valuable insights. Promotions and deals can be created and managed to attract customers and boost sales effectively.

Super Admin Functionality:

Super admins have overarching control over the platform, with functionalities such as platform management, user account management, and maintaining system performance. Content management tools enable super admins to manage banners and promotional content displayed on the platform effectively. Performance monitoring functionalities allow super admins to analyze platform usage data, identify trends, and ensure smooth system operation.

LIST OF FIGURES

FIGURE	NAME	PAGE NO.
Fig 3.1	HTML	05
Fig 3.2	CSS	06
Fig 3.3	JavaScript	06
Fig 3.4	React Js	07
Fig 3.5	Bootstrap	09
Fig 3.6	Mongo DB	09
Fig 3.7	Node js	10
Fig 3.8	Express js	11
Fig 4.1	Visual Studio Code	12
Fig 6.1.1	PWA Flow Chart	16
Fig 6.1.2	Scanner Screen	17
Fig 6.1.3	Menu Screen	17
Fig 6.1.4	Login	17
Fig 6.1.5	Selected Order	17
Fig 6.1.6	Search Box	18
Fig 6.1.7	My orders.	18
Fig 6.1.8	Login During order	18
Fig6.1.9	Order Successful	18
Fig 6.1.10	Order Status	19
Fig 6.1.11	Order History	19
Fig 6.1.12	Order Details	19
Fig 6.1.13	Edit Profile	19
Fig 6.2	Admin Flow Chart	20
Fig 6.2.1	Admin login	21
Fig6.2.2	Forget Password Page	21
Fig6.2.3	Dashboard	22
Fig6.2.4	Profile	22
Fig 6.2.5	Edit Profile	23
Fig 6.2.6	Order Status	23
Fig 6.2.7	Notifications	23
Fig 6.2.8	Orders	24

Fig 6.2.9	Order Filters	24
Fig 6.2.10	Add Items	25
Fig 6.2.11	Edit Item	25
Fig 6.2.12	Add Quantity	26
Fig 6.2.13	Edit Quantity	26
Fig 6.2.14	Add Miscellaneous	26
Fig 6.2.15	Edit Miscellaneous	26
Fig 6.2.16	Add Category	27
Fig 6.2.17	Edit Category	27
Fig 6.2.18	Delete Category	27
Fig 6.2.19	Filter by Category	27
Fig 6.2.20	Filter by Search	28
Fig 6.2.21	Add/Edit Taxes	28
Fig 6.2.22	QR Codes	29
Fig 6.2.23	Print QR Code	29
Fig 6.2.24	Folder Structure for Backend	30

TABLE OF CONTENT

S.NO.	TOPIC	PAGE NO.
	<i>Internship Certificate</i> <i>Declaration</i> <i>Acknowledgement</i> <i>Abstract</i> <i>List of Figures</i>	<i>i</i> <i>ii</i> <i>iii</i> <i>iv</i> <i>v-vi</i>
1.	CHAPTER - 1 INTRODUCTION TO ORGANIZATION	1-2
2.	CHAPTER - 2 INTRODUCTION TO PROJECT 1.1 Project Overview 1.2 Problem Statement 1.3 Objectives 1.4 Scope of the project 1.5 Significance of the project	3-4
3.	CHAPTER - 3 TECHNOLOGIES USED 2.1 Html 2.2 CSS 2.3 JavaScript 2.4 ReactJS 2.5 Bootstrap 2.6 MongoDB 2.7 Node Js	5-11

	2.8 Express Js	
3.	CHAPTER – 4 TOOLS USED	12-13
4.	CHAPTER – 5 METHODOLOGY	14-15
5.	CHAPTER – 6 DETAILS OF PROJECT 6.1 Front End 6.2 Backend	16-32
6.	CHAPTER – 7 RESULTS AND DISCUSSIONS	33-34
7.	CHAPTER – 8 CONCLUSION AND FUTURE SCOPE	35-36
	BIBILOGRAPHY	37

CHAPTER - 1

INTRODUCTION TO ORGANIZATION

Sensation Solutions is one of the best Companies, well defined to offer customer centric B2B, B2C services & indigenous products for various industry verticals such as Digital Services, IT Services, Gaming, Travel, Business Consultancy & Legal Services. We have a team of experts that leaves no stone unturned to provide the best solutions.

In the year 2013, Sensation Solutions was born with the aim of empowering local businesses to expand and register their presence globally. With each passing year, we grew exponentially and the reason behind the tremendous growth, To the point process, our customer interaction, Development Strategies and of course our dedication to build something amazing par excellence. Today we have reached the stage we are in stands in league of top market leaders in the IT & Digital Services world that specializes in providing the top-notch services. You name any IT service; we are able to provide it.

- We believe in **understanding the requirements** of our clients in order to develop a perfect solution which can solves their business problems.
- Our budget-friendly approach always makes our client relaxed.
- Getting a project done with full **Confidentiality** has been always a top priority for us.
- We have gained appreciation and trust of our customers due to our end to end services under a roof.

Our Value

We aspire to transform into market leader in helping the business through digital means. Client satisfaction, transparency, teamwork and growth are the core values behind Sensation Solutions that reflects who we are, how we work and what we endeavor.



Our Vision

Our vision is to give growth to all those who are associated with us. Grow our customers by providing them with the best IT services that drive their business forward. Grow our employees through their all round grooming. Company's growth by managing everything in a good manner.



Our Mission

Focused on our Vision, we are aggressively working on achieving it. We are continuously expanding our client base, enhancing our skillset and applying best of brains to manage the things.



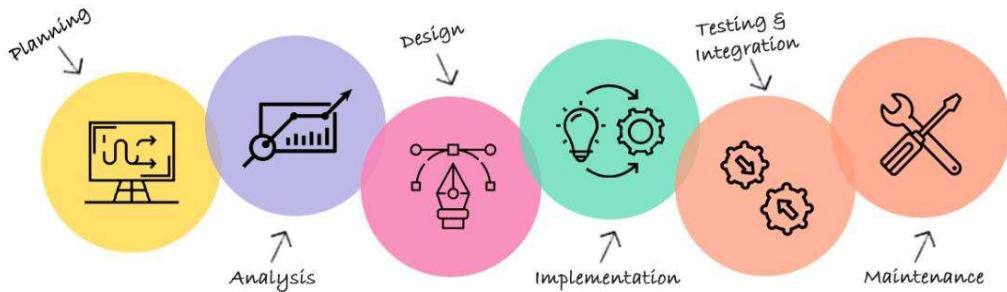
Our Strategy

Our strategy to achieve our mission is to foresee the situation and get planned for that. It helps us to keep going and be prepared with the solution for the obstacles to achieve the goal. Ultimately it gives a smooth growth to all who are associated with us.

Methodology

We follow Agile methodology by thoroughly analyzing and understanding your business requirements to help you take maximum advantage of the latest technologies. We focus on aesthetic factors like regular scrum meetings, planning, designing, developing, testing and

launching. We use best methodologies and approaches when it comes to web development. It is important to assess the scale and aspects of your projects in order to select the best approach every time.



Organization Clients



Services Provided by Organization

Services

DEVELOPMENT

- Web Development
- Software Development
- Mobile App Development
- E-commerce Development
- Product Development
- Social Media App Development
- CMS Development
- Database Management
- Online Storefront Builder

UI & UX DESIGN

- Brand designing
- Web & Mobile App Design
- Digital Design
- Social Media Design

QUALITY ASSURANCE

- Automation Testing
- Manual Testing

CHAPTER – 2

INTRODUCTION TO PROJECT

1.1 Project Overview:

The Restaurant Order Management Project by Sensation Software Solutions Pvt Ltd aims to modernize the dining experience through a web-based platform. Utilizing cutting-edge technologies, it facilitates seamless interactions between customers, restaurants, and super admins. This project addresses the evolving needs of the food service industry by providing intuitive functionalities for efficient order management, personalized customer experiences, and comprehensive platform administration.

1.2 Problem Statement:

Traditional restaurant operations often suffer from inefficiencies in order processing, customer engagement, and platform management. Manual order taking, limited customer insights, and inadequate administrative tools hinder productivity and hinder growth potential. The need for a modernized system that streamlines operations, enhances customer satisfaction, and empowers platform administrators is evident in the increasingly competitive restaurant landscape.

1.3 Objectives:

1. To streamline the ordering process for customers by providing a user-friendly interface for browsing menus, customizing orders, and leaving reviews, thus enhancing the overall dining experience.
2. To empower restaurants with efficient management tools for updating menus, processing orders, generating invoices, and analyzing performance, thereby optimizing operational efficiency and driving revenue growth.
3. To empower super admins with comprehensive platform management functionalities, including user account management, content management, performance monitoring, and security enforcement, ensuring smooth system operation and regulatory compliance.

1.4 Scope of the Project:

The scope of the Restaurant Order Management Project encompasses the development of a web-based platform that caters to three main user roles: customers, restaurants, and super admins. Key functionalities include seamless ordering for customers, comprehensive restaurant management tools, and robust platform administration features. The project involves utilizing modern technologies such as HTML, CSS, JavaScript, React, MongoDB, Express, and Node.js to create an intuitive and efficient system. The scope also includes implementing features such as user account

creation, menu management, order processing, billing and analytics, promotions and deals management, content management, performance monitoring, and security measures.

1.5 Significance of the Project:

The Restaurant Order Management Project holds significant importance for various stakeholders in the food service industry. For customers, it offers a modernized and convenient dining experience, allowing for seamless ordering, personalized recommendations, and easy access to restaurant information and reviews. For restaurants, the project provides essential management tools for efficiently processing orders, managing menus, tracking sales performance, and implementing promotional activities to attract and retain customers. Super admins benefit from comprehensive platform management functionalities, enabling them to oversee operations, analyze performance, and ensure regulatory compliance. Overall, the project aims to streamline restaurant operations, enhance customer satisfaction, and drive business growth in an increasingly competitive industry landscape. Its significance lies in its potential to revolutionize the way restaurants interact with customers and manage their operations, ultimately leading to improved efficiency, profitability, and customer loyalty.

CHAPTER - 3

TECHNOLOGY USED

Front-End Technologies

3.1 HTML

HTML, or Hypertext Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It provides the basic structure and content of web pages, defining elements such as headings, paragraphs, links, images, and more.

HTML uses a system of tags to define the structure of a web page. Tags are enclosed in angle brackets (<>) and typically come in pairs, with an opening tag (e.g., <tag>) and a closing tag (e.g., </tag>). The content of a web page is placed between these tags, and each tag provides specific instructions to the web browser on how to display the content.

In addition to defining the structure of a web page, HTML also allows for the inclusion of other resources such as images, videos, and scripts, which enhance the functionality and interactivity of the page. Combined with CSS (Cascading Style Sheets) for styling and JavaScript for interactivity, HTML forms the foundation of the modern web.



Fig- 3.1

3.2 CSS

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML, etc.). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS makes it easy to control the layout, style, and appearance of multiple web pages all at once. By separating the presentation of a document from its structure, CSS simplifies the process of creating visually engaging web pages and ensures consistency across a website.

CSS uses a simple syntax that consists of selectors, properties, and values. Selectors are used to target specific elements in an HTML document, while properties define the style aspects (such as

color, font, size, etc.) of those elements. Values are assigned to properties to specify the desired style.

With CSS, web developers can create responsive and attractive web designs that adapt to different screen sizes and devices. CSS also allows for the creation of animations and other dynamic effects, enhancing the user experience on the web.



Fig- 3.2

3.3 JavaScript

JavaScript is a versatile programming language primarily used for creating interactive and dynamic content on websites. It is a key technology for web development, allowing developers to add functionality, interactivity, and behavior to web pages.

Originally developed by Netscape, JavaScript is now supported by all modern web browsers and can be used on both the client side (in the browser) and the server side (with Node.js). It is known for its flexibility and ease of use, making it an ideal choice for web development.

JavaScript is used to manipulate the Document Object Model (DOM) of a web page, enabling developers to dynamically change the content, structure, and style of a page in response to user actions or other events. It can also be used to create animations, validate form data, and interact with web services through APIs.

Overall, JavaScript plays a crucial role in modern web development, allowing developers to create rich, interactive web applications that enhance the user experience.



Fig- 3.3

3.4 React JS

React is a powerful JavaScript library for building user interfaces, developed by Facebook. It has gained immense popularity among developers for its simplicity, flexibility, and efficiency. React allows you to create interactive and dynamic UI components, which can be easily reused across your application.

One of the key features of React is its use of a virtual DOM (Document Object Model), which enables it to efficiently update and render components based on changes in data, without reloading the entire page. This makes React applications fast and responsive, providing a smooth user experience.

React also follows a component-based architecture, where UIs are divided into reusable components, making the codebase more modular and easier to maintain. Additionally, React's declarative approach to building UIs allows developers to describe how the UI should look based on the current application state, rather than manually manipulating the DOM.

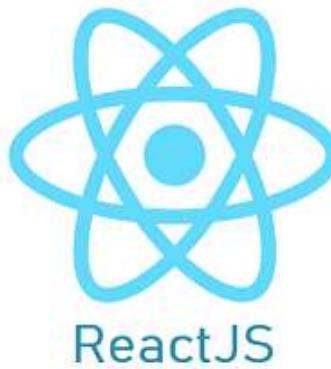


Fig- 3.4

Advantages:

React has several advantages that make it a popular choice for building user interfaces:

1. Declarative:

React makes it easy to build interactive UIs by using a declarative programming style. Developers describe how the UI should look based on the current application state, and React handles updating the DOM to match this state.

2. Component-Based:

React uses a component-based architecture, where UIs are divided into reusable components. This makes the codebase more modular and easier to maintain, as components can be reused across the application or in different projects.

3. Virtual DOM:

React uses a virtual DOM to keep track of changes to the UI. This allows React to update the DOM efficiently by only updating the parts of the DOM that have changed, rather than re-rendering the entire DOM.

4. Efficient Updates:

Because React uses the virtual DOM, it can batch updates and minimize DOM manipulations, resulting in better performance compared to traditional DOM manipulation techniques.

5. One-Way Data Binding:

React uses a one-way data flow, where data flows downwards from parent components to child components. This makes it easier to understand how data is passed between components, and helps prevent bugs caused by two-way data binding.

6. JSX:

React uses JSX, a syntax extension that allows you to write HTML-like code within your JavaScript. This makes it easier to write and maintain UI components, as you can use familiar HTML syntax.

7. Community and Ecosystem:

React has a large and active community, with a wealth of resources and libraries available to help you build your application. This includes tools for state management, routing, and testing, among others.

8. Performance:

React's virtual DOM and efficient update mechanism make it well-suited for building high-performance web applications. It is particularly well-suited for applications with complex UIs that require frequent updates

9. Reusable Components:

React's component-based architecture allows developers to create reusable UI components. These components can be used across different parts of the application, promoting code reusability and reducing development time.

10. Unidirectional Data Flow:

React follows a unidirectional data flow, where data flows in a single direction from parent components to child components. This makes it easier to manage and debug data changes, as each component's state is isolated and predictable.

11. Virtual DOM Reconciliation:

React's virtual DOM allows for efficient updates to the DOM by only re-rendering components that have changed. This minimizes the number of DOM manipulations,

resulting in improved performance.

12. Developer Tools:

React provides a set of developer tools that make it easier to debug and inspect React components. These tools allow developers to track component hierarchies, inspect component props and state, and analyze performance.

3.5 Bootstrap

Bootstrap is a popular open-source front-end framework for developing responsive and mobile-first websites. It provides a collection of CSS and JavaScript tools, including pre-designed components like buttons, forms, navigation bars, and modals, that help developers create modern and visually appealing web interfaces quickly and efficiently.



Fig- 3.5

3.6 Mongo DB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it ideal for handling large volumes of unstructured or semi-structured data. It allows for horizontal scaling and offers high performance and availability. MongoDB's document model is easy to learn and use, making it popular for developing applications that require fast iteration and diverse data handling.



Fig- 3.6

Advantages:

1. Flexible Schema:

Unlike traditional relational databases, MongoDB does not require a predefined schema. This flexibility allows you to easily change the structure of your documents as your application evolves, without needing to modify the entire database schema.

2. Scalability:

MongoDB is designed to scale horizontally across multiple servers, allowing you to handle large volumes of data and high traffic loads. It supports sharding, which distributes data across multiple servers, and replication, which provides high availability and fault tolerance.

3. Performance:

MongoDB offers high performance for read and write operations, thanks to its efficient indexing, query optimization, and in-memory storage engine. It also supports operations like MapReduce and aggregation pipelines for complex data processing.

4. Rich Query Language:

MongoDB provides a powerful query language that supports a wide range of operations, including CRUD operations (Create, Read, Update, Delete), text search, geospatial queries, and more.

5. Ad Hoc Queries:

MongoDB allows you to perform ad hoc queries on your data without needing to define a schema or create complex joins, making it easy to explore and analyze your data.

6. Community and Ecosystem:

MongoDB has a large and active community of developers, who contribute to its development and provide support through forums, meetups, and online resources. It also has a rich ecosystem of tools and libraries for integrating MongoDB with various programming languages and frameworks.

3.7 Node Js

Node.js is a JavaScript runtime built on Chrome's V8 engine. It enables server-side scripting and allows developers to use JavaScript to write backend code, making it possible to create scalable and efficient network applications. Node.js is known for its non-blocking, event-driven architecture, which makes it well-suited for real-time applications like chat services and online gaming.



Fig- 3.7

3.8 Express Js

Express.js is a minimalist web application framework for Node.js. It provides a robust set of features to build single-page, multi-page, and hybrid web applications. Express simplifies the process of handling HTTP requests, managing middleware, and rendering views, making it a fundamental tool for creating server-side applications in a Node.js environment.



Fig- 3.8

CHAPTER - 4

TOOLS USED

Visual Studio Code

Visual Studio Code (VS Code) is a lightweight but powerful source code editor developed by Microsoft for Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript, and Node.js, and has a rich ecosystem of extensions for other languages (such as C++, C#, Python, and more) and functionalities.

VS Code provides a modern and intuitive user interface, with features like IntelliSense (code completion), debugging support, Git integration, and a built-in terminal. It also offers customizable themes, keyboard shortcuts, and settings, allowing developers to tailor the editor to their preferences and workflow.

With its fast performance, extensibility, and wide range of features, Visual Studio Code has become one of the most popular code editors among developers for various programming tasks, from web development to cloud applications.



Fig- 4.1

Advantages:

Visual Studio Code (VS Code) has several advantages that make it a popular choice among developers:

- Cross-Platform:** VS Code is available for Windows, macOS, and Linux, allowing developers to use the same editor across different operating systems.
- Free and Open Source:** VS Code is free to download and use, and its source code is available on GitHub, allowing developers to contribute to its development and customize it to their needs.

3. **Lightweight:** Despite its powerful features, VS Code is lightweight and fast, making it suitable for use on low-spec machines.
4. **Rich Extension Ecosystem:** VS Code has a rich ecosystem of extensions that add functionality for different languages, frameworks, and tools. These extensions enhance the editor's capabilities and can be easily installed from the VS Code Marketplace.
5. **IntelliSense:** VS Code provides IntelliSense, which offers intelligent code completion, code navigation, and context-aware suggestions, making coding faster and more efficient.
6. **Built-in Git Integration:** VS Code has built-in Git integration, allowing developers to easily manage source code repositories, commit changes, and resolve conflicts without leaving the editor.
7. **Debugging Support:** VS Code provides built-in debugging support for various languages and frameworks, allowing developers to debug their code directly within the editor.
8. **Customizable:** VS Code is highly customizable, with support for themes, keyboard shortcuts, and settings. Developers can personalize the editor to match their preferences and workflow.
9. **Active Community and Support:** VS Code has a large and active community of developers, with a wealth of documentation, tutorials, and forums available for support and assistance.

CHAPTER-5 **METHODOLOGY**

Project Planning and Requirements Gathering:

- Conduct thorough discussions with stakeholders (Sensation Software Solutions Pvt Ltd, potential users, etc.) to understand project objectives, scope, and requirements.
- Define user stories and use cases to capture functionalities required for customers, restaurants, and super admins
- Create a project plan outlining timelines, milestones, and resources needed for development.

Technology Selection and Setup:

- Evaluate and select appropriate technologies for front-end (HTML, CSS, JavaScript, React) and back-end (Node.js, Express, MongoDB) development based on project requirements and team expertise.
- Set up development environments and necessary tools such as version control (e.g., Git), project management (e.g., Jira), and communication (e.g., Slack).

System Design and Architecture:

- Design the overall system architecture considering factors like scalability, performance, and security.
- Define database schema for storing customer, restaurant, and order-related information using MongoDB.
- Design RESTful APIs for communication between front-end and back-end components.

Frontend Development:

- Develop user interfaces for customers, restaurants, and super admins using HTML, CSS, JavaScript, and React.
- Implement features such as browsing menus, placing orders, user account management, and review/rating systems.
- Ensure responsive design for compatibility across various devices and browsers.

Backend Development:

- Develop server-side logic using Node.js and Express to handle requests from the front-end.
- Implement authentication and authorization mechanisms for secure access to different functionalities.
- Build APIs for managing restaurant information, menus, orders, billing, analytics, etc.
- Integrate with MongoDB for data storage and retrieval.

Testing:

- Conduct unit tests, integration tests, and end-to-end tests to ensure the reliability and functionality of the system.
- Perform usability testing to validate user flows and interfaces.
- Address any issues or bugs identified during testing iterations.

Deployment and Launch:

- Deploy the web application to a hosting environment (e.g., AWS, Heroku) ensuring scalability and reliability.
- Configure domain, SSL certificates, and other necessary settings for production deployment.
- Perform final checks to ensure everything is functioning as expected before the official launch.

Post-launch Support and Maintenance:

- Provide ongoing support to address user queries, issues, and feedback.
- Monitor system performance, security, and scalability, making necessary optimizations and updates.
- Continuously iterate on the platform based on user feedback and evolving requirements.

CHAPTER-6

DETAILS OF PROJECT

Frontend of the Project

6.1 Workflow of Progressive Web App (PWA) and Screenshots

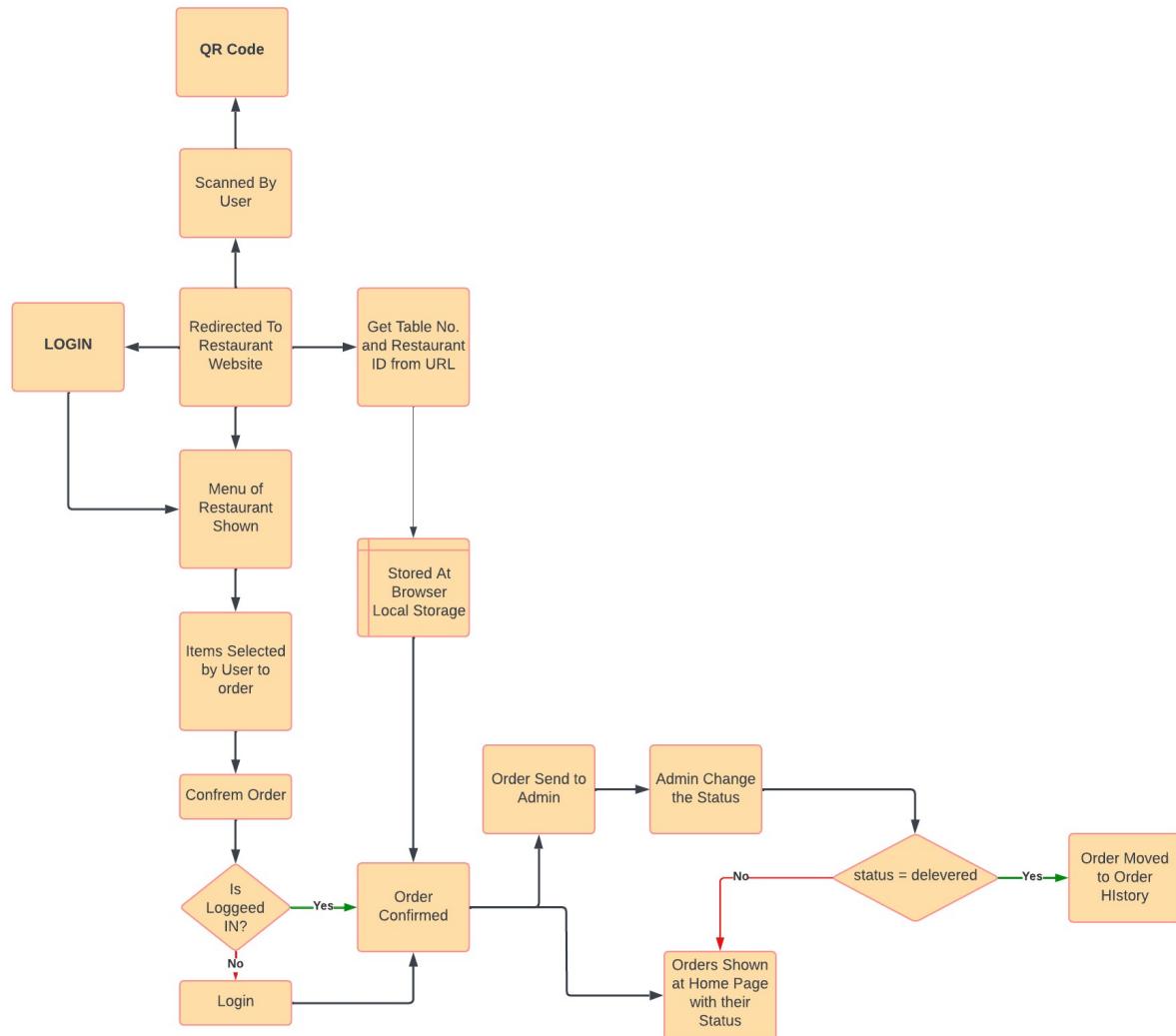


Fig:6.1.1 Flow Chart

Screenshots of Menu App

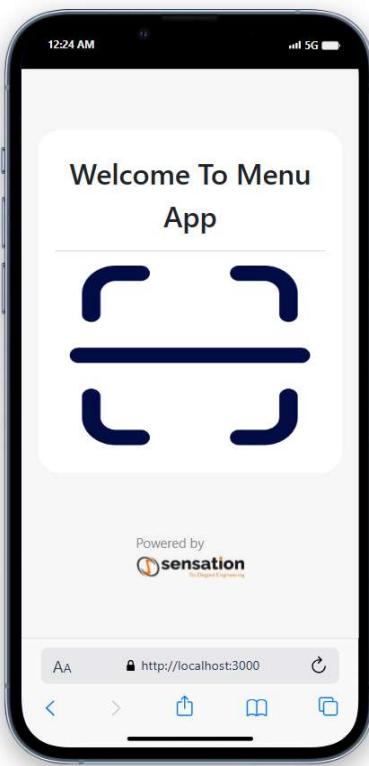


Fig 6.1.2 Scanner Screen

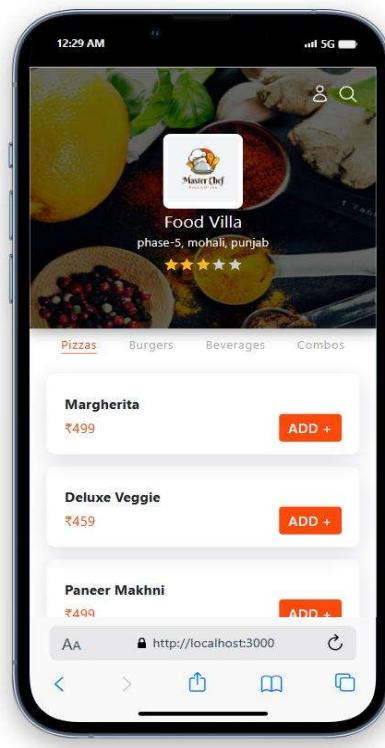


Fig 6.1.3 Menu Screen

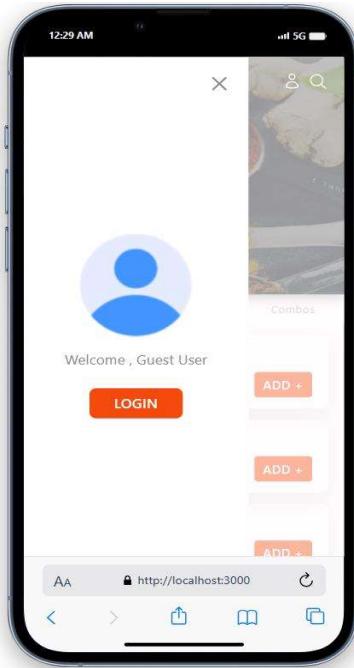


Fig 6.1.4 Login

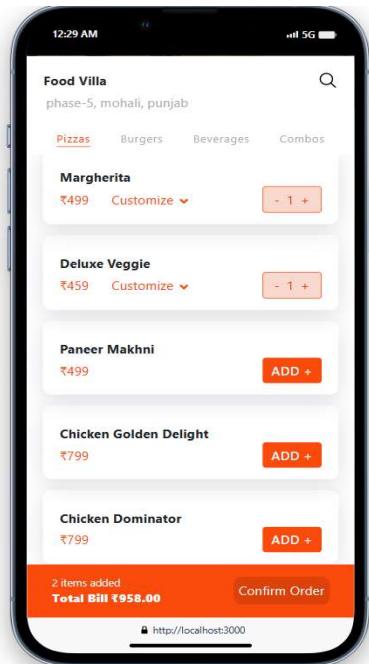


Fig 6.1.5 Selected Order

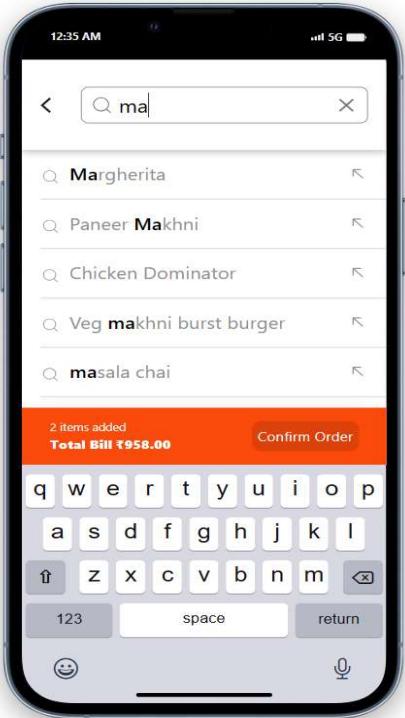


Fig 6.1.6 Search Box

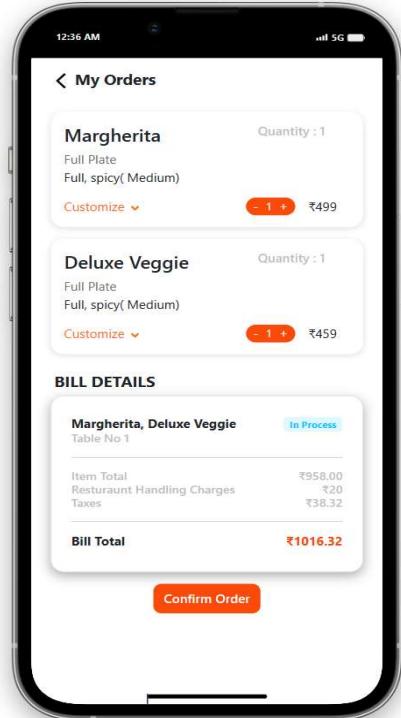


Fig 6.1.7 My Order

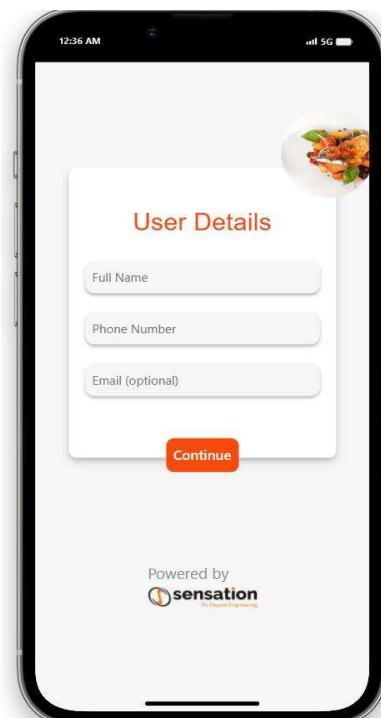


Fig 6.1.8 Login During Order

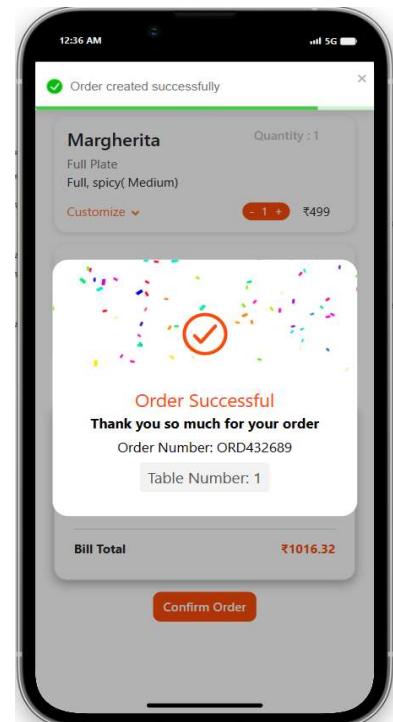


Fig 6.1.9 Order Successful

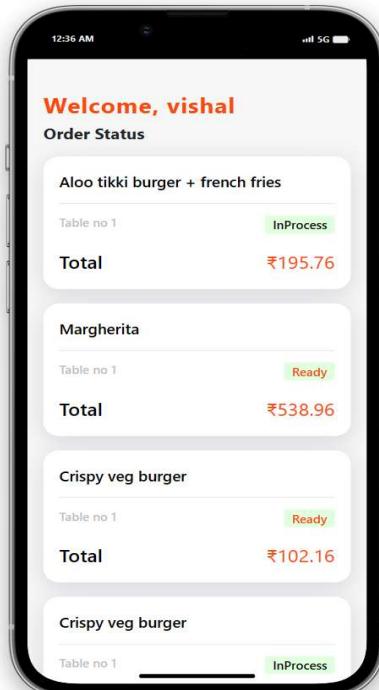


Fig 6.1.10 Order Status

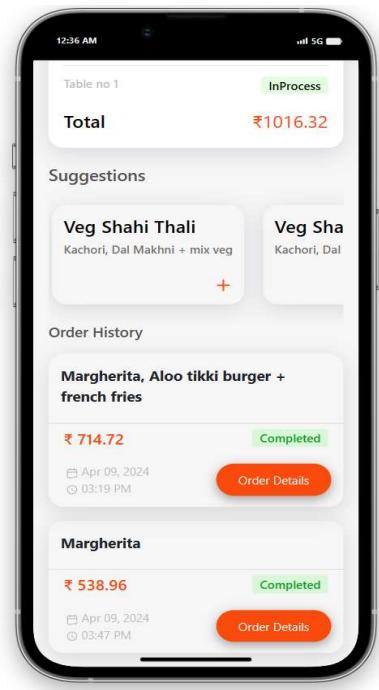


Fig 6.1.11 Order History

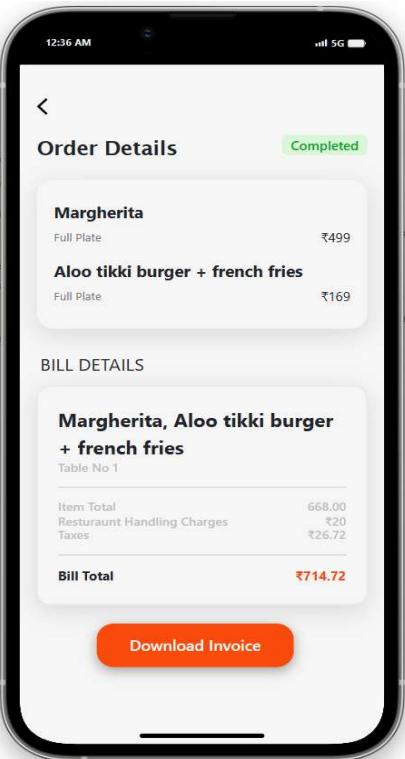


Fig 6.1.12 Order Details

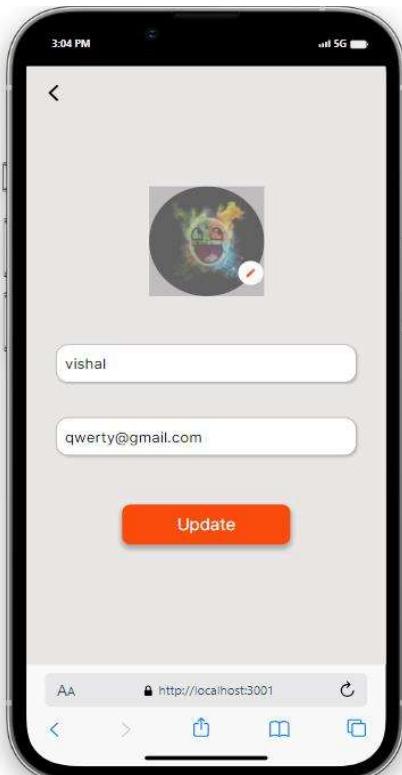


Fig 6.1.13 Edit Profile

6.2 Workflow of Restaurant Side and Screenshots

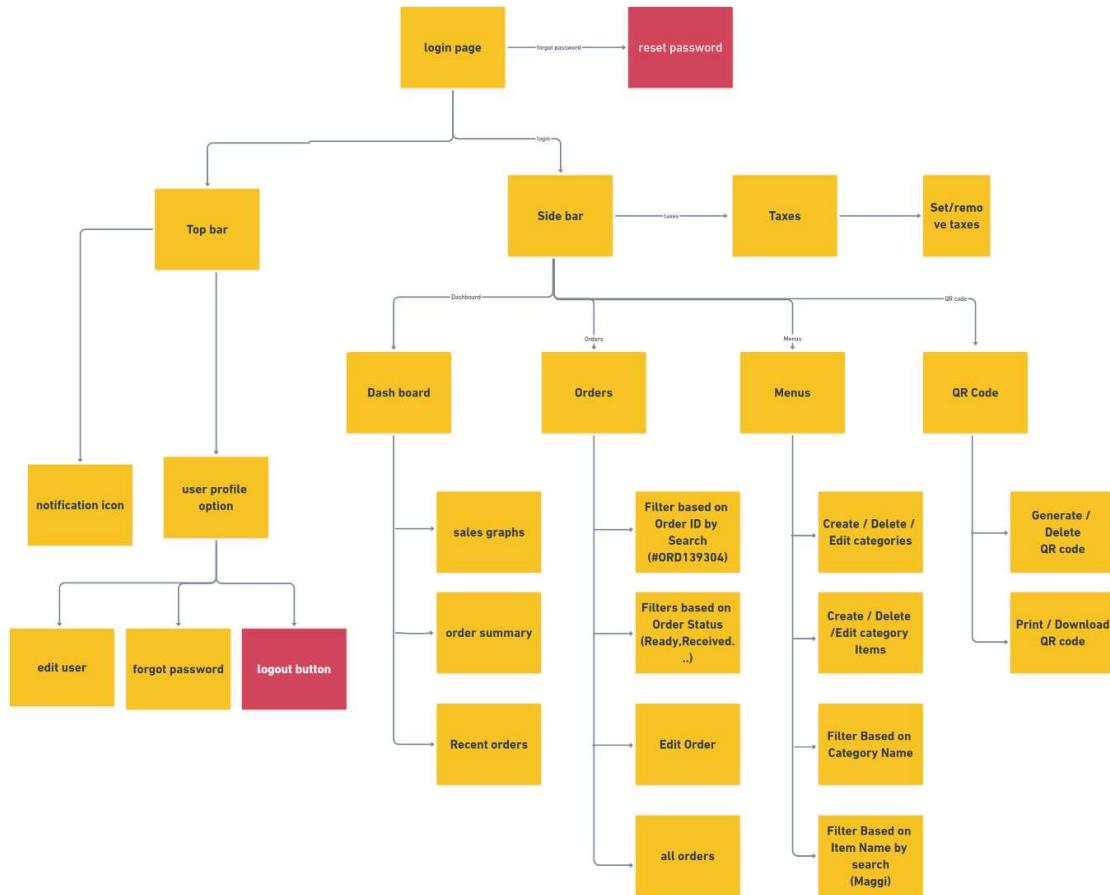


Fig:6.2 Flow Chart of Admin

Screenshots of Admin (Restaurant Side)

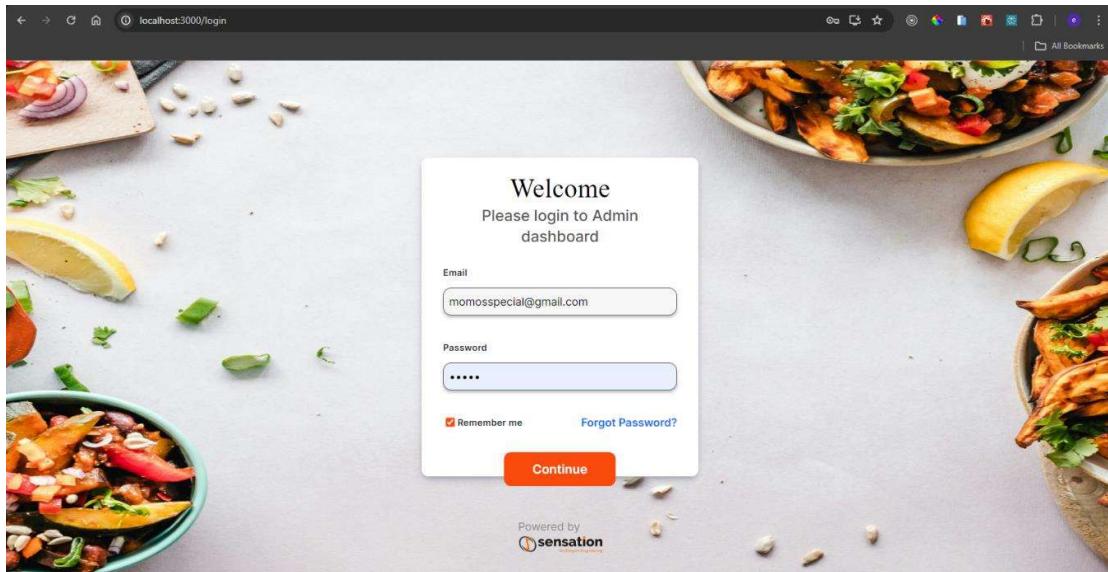


Fig 6.2.1 Login Page

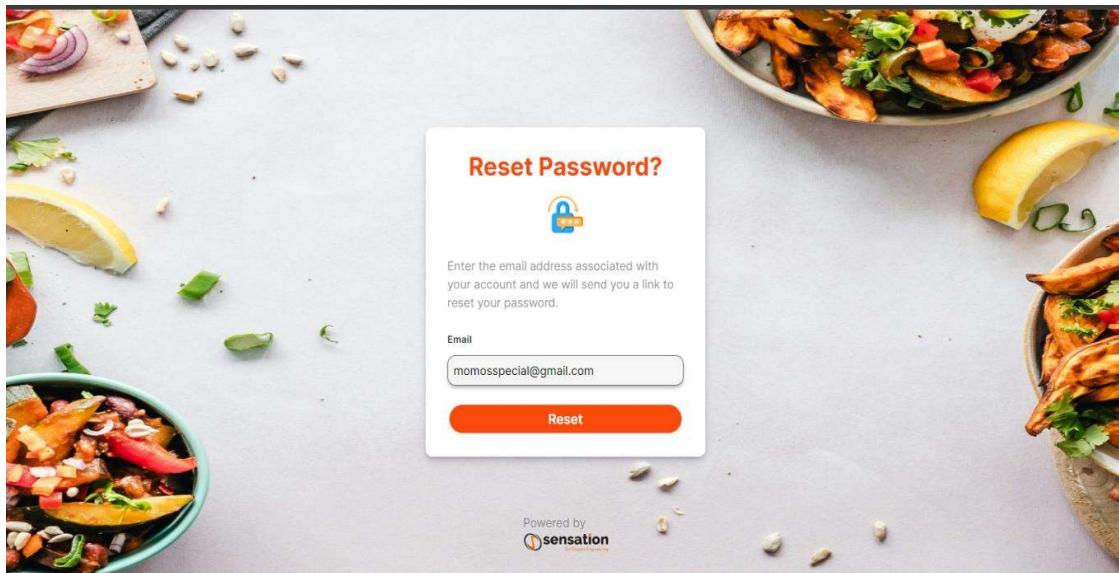


Fig 6.2.2 Forget Password Page

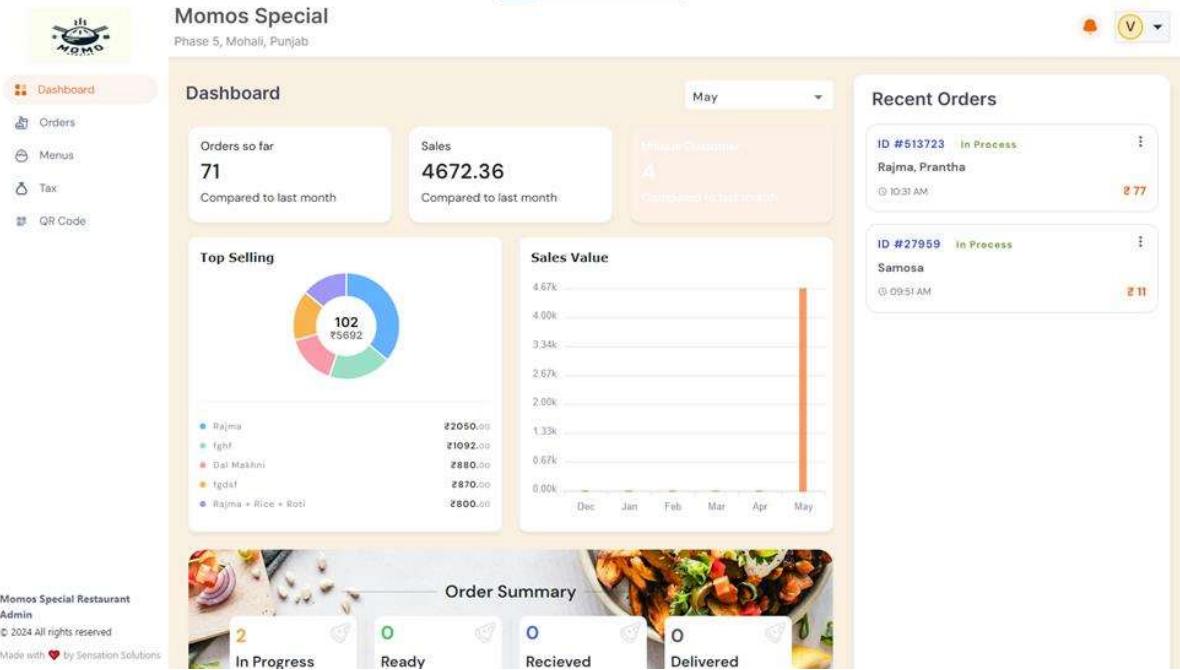


Fig 6.2.3 Dashboard

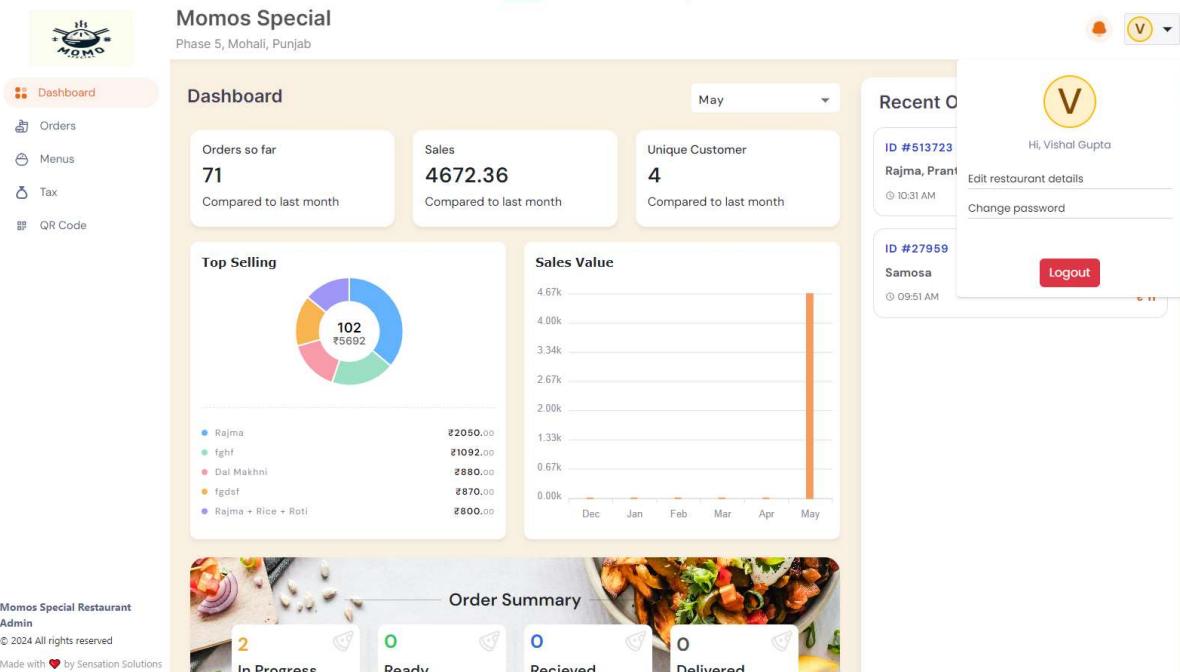


Fig 6.2.4 Profile

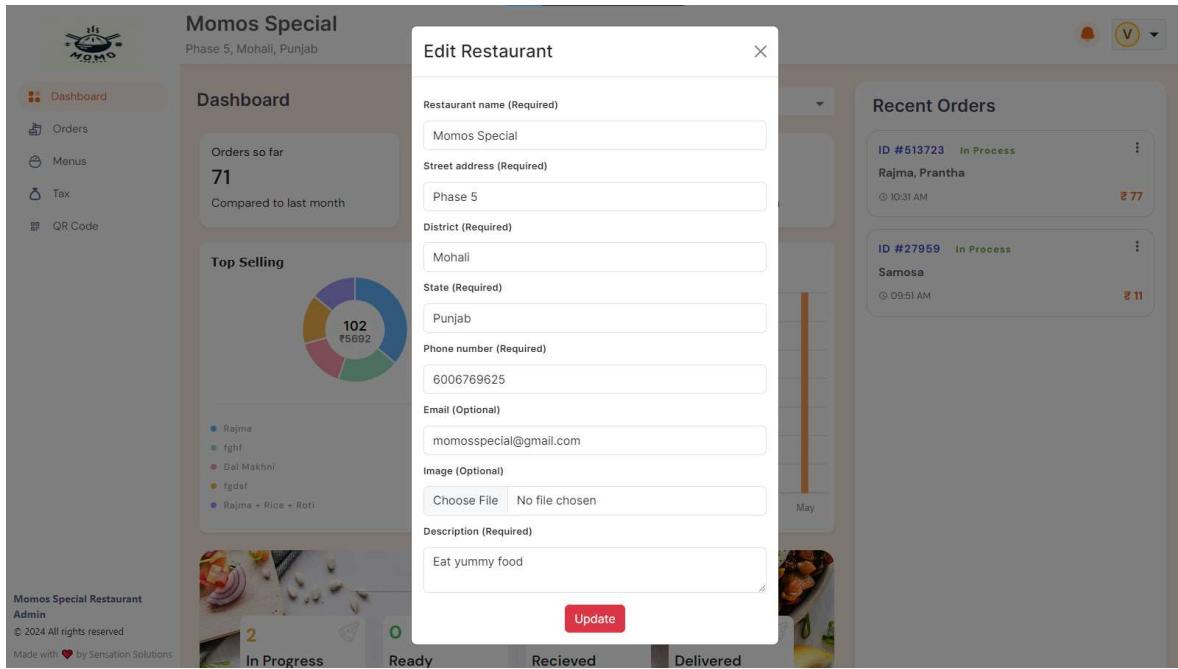


Fig 6.2.5 Edit Profile

Recent Orders

- ID #513723 In Process Rajma, Prantha @ 10:31 AM
- ID #27959 In Pro Samosa @ 09:51 AM

Notifications

No notifications yet

Fig 6.2.6 Order Status

Fig 6.2.7 Notifications

Momos Special

Phase 5, Mohali, Punjab

Food Orders 😊

+ Create Order

Search

Enter Order ID to Search

All 102 Ready 3 Received 3 More Filters

ID : ORD879706
vishal Table No : 1
qwerty@gmail.com +91-6006769602 In Process

ID : ORD879706
vishal Table No : 1
qwerty@gmail.com +91-6006769602 In Process

ID #ORD979633 • InProcess
Rajma 09:49 AM ₹52.5

ID #ORD970351 • delivered
Rajma 02:20 PM ₹52

ID #ORD949008 • InProcess
Rajma 12:21 PM ₹52

ID #ORD918262 • received
Dal Makhni + Dal Makhni + Dal Makhni 12:17 PM ₹166.4

ID #ORD101613 • received
Rajma + Rajma 11:35 AM ₹156

ID #ORD647429 • delivered
Coke 10:46 AM ₹20.8

ID #ORD60005 • delivered
Rajma + Rice + Roti 10:36 AM ₹520

ID #ORD829483 • InProcess
Rajma 06:23 PM ₹21.0

Order items

Rajma ✓ Half - 1 + ₹50

Item Total ₹50
Taxes ₹9.00
Bill Total ₹59

Momos Special Restaurant Admin © 2024 All rights reserved Made with ❤ by Sensation Solutions

Fig 6.2.8 Orders

Food Orders 😊

+ Create Order

Search

Enter Order ID to Search

All 102 Ready 3 Received 3 More Filters

Rajma + Dal Makhni + Prantula

Filters

Delivered 71
Preparing 23
Cancelled 2
Price Range: ₹0 – ₹10000

ID #ORD838944 • received
Samosa + Manchurian

ID #ORD930928 • InProcess
Rajma + Dal Makhni

ID #ORD969038 • InProcess
Rajma + Dal Makhni 11:45 AM ₹93.6

ID #ORD980571 • cancelled
Rajma + Dal Makhni 11:44 AM ₹93.6

ID #ORD675060 • InProcess
Rajma 11:35 AM ₹0

ID #ORD626319 • InProcess
Rajma + Rice + Roti 05:27 PM ₹104

ID #ORD356545 • InProcess
Gulab Jamun 04:42 PM ₹41.6

Fig 6.2.9 Order Filters

Momos Special
Phase 5, Mohali, Punjab

Dashboard Orders Menus Tax QR Code

Menu List

Type Menu Item Filter

- Food(9)
 - Rajma ₹ 50
 - Dal Makhni ₹ 40
 - Prantha ₹ 20
 - Shahi paneer ₹ 234
 - Aloo matar ₹ 35
- + Add Sub Category
- + Add Item
- Deserts(4)
- Drinks(4)
- Snacks(3)
- Thali Special(3)
- + Add Menu Category

Add Item

Item Name: Rajma Category: Food

Quantity: Half ₹50 Full ₹100

Description: it is one of the famous dish to j6k

Food Type: Veg Egg Non-Veg

Miscellaneous: Less spicy High Spicy

Close Save

Fig 6.2.10 Menu page Add Item

Momos Special
Phase 5, Mohali, Punjab

Dashboard Orders Menus Tax QR Code

Menu List

Type Menu Item Filter

- Food(9)
 - Rajma ₹ 50
 - Dal Makhni ₹ 40
 - Prantha ₹ 20
 - Shahi paneer ₹ 234
 - Aloo matar ₹ 35
- + Add Sub Category
- + Add Item
- Deserts(4)
- Drinks(4)
- Snacks(3)
- Thali Special(3)
- + Add Menu Category

Edit Item

Item Name: Rajma Category: Food

Quantity: Half ₹50 Full ₹100

Description: it is one of the famous dish to j6k

Food Type: Veg Egg Non-Veg

Miscellaneous: Less spicy High Spicy

Close Update

Fig 6.2.11 Edit Item

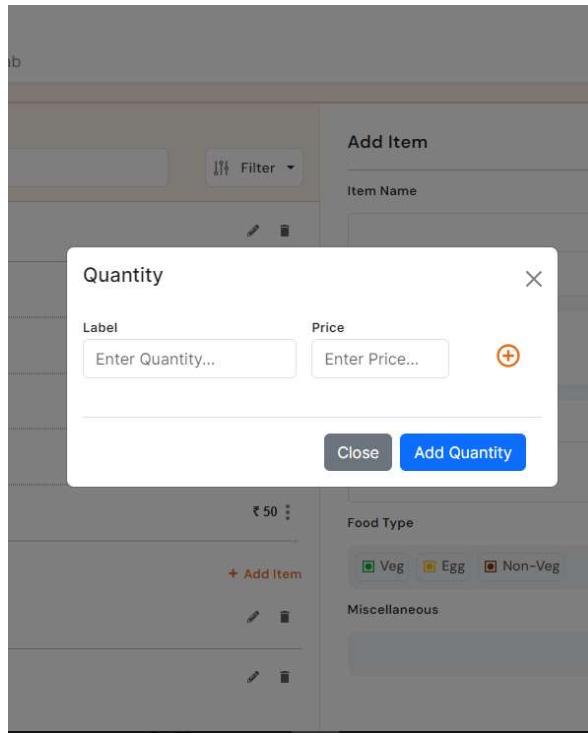


Fig 6.2.12 Add Quantity

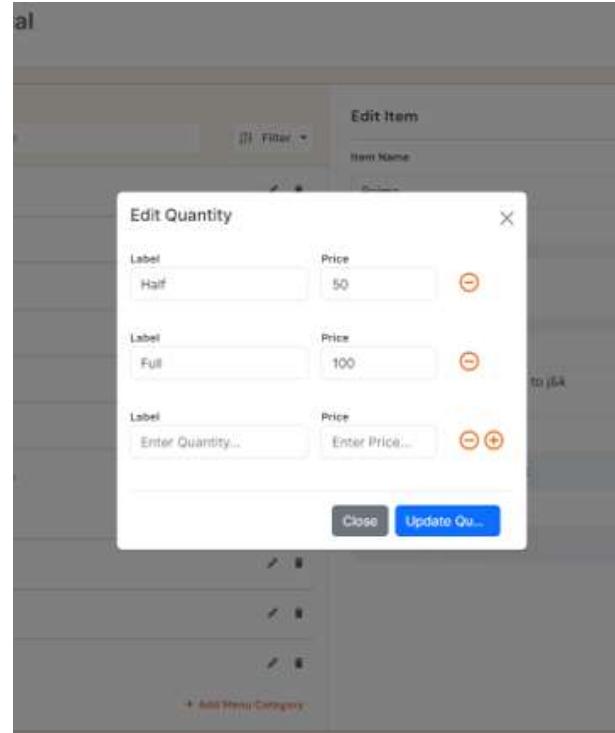


Fig 6.2.13 Edit Quantity

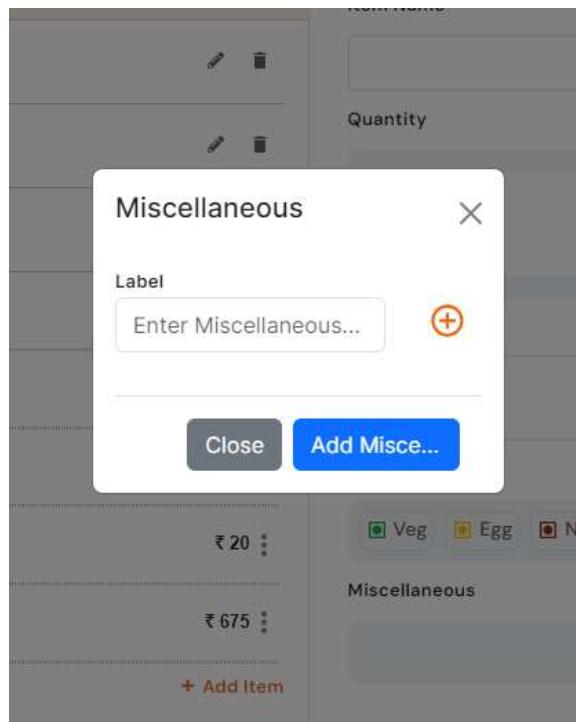


Fig 6.2.14 Add Miscellaneous

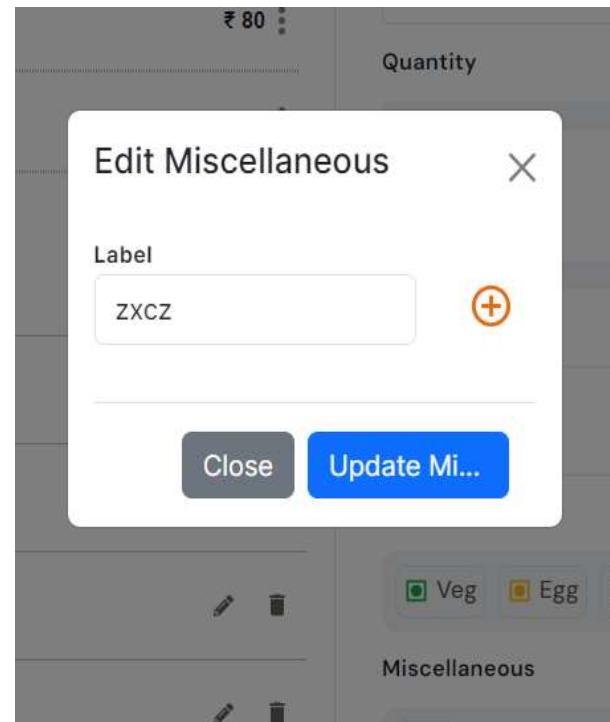


Fig 6.2.15 Edit Miscellaneous

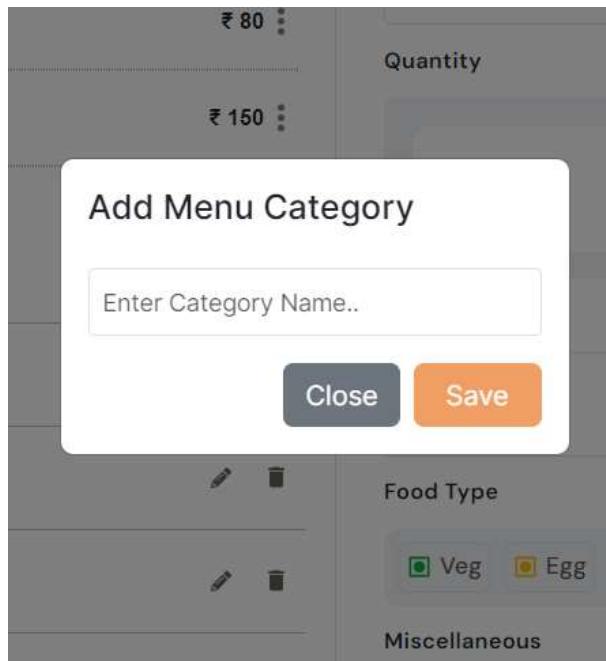


Fig 6.2.16 Add Category

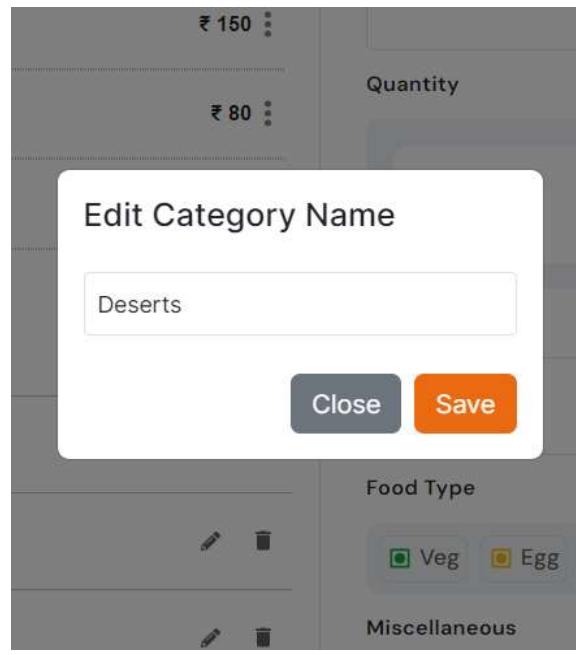


Fig 6.2.17 Edit Category

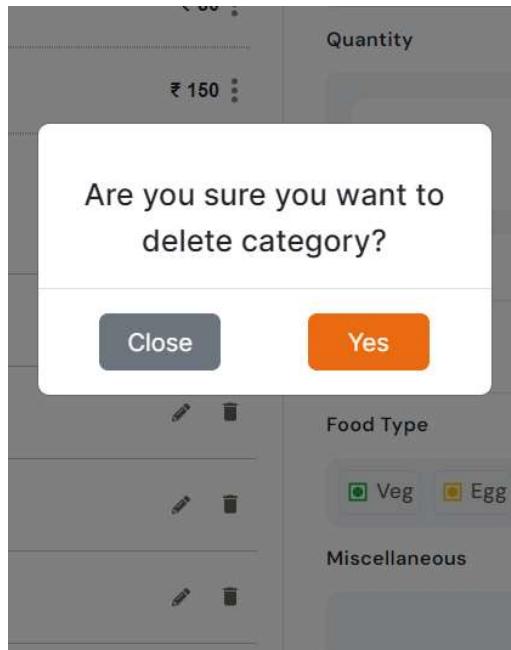


Fig 6.2.18 Delete Category

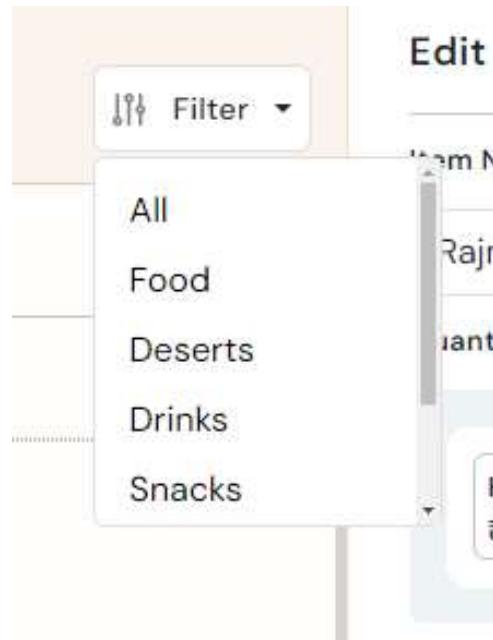


Fig 6.2.19 Filter by Category

Momos Special
Phase 5, Mohali, Punjab

Dashboard Orders Menus Tax QR Code

Menu List

Search: raj Filter ▾

Food(1)

Rajma ₹ 50

+ Add Sub Category + Add Item

Deserts(0)

Drinks(0)

Snacks(0)

Thali Special(1)

Rajma + Rice + Roti ₹ 100

+ Add Menu Category

Edit Item

Item Name: Rajma Category: Food

Quantity: Half ₹50 Full ₹100

Description: it is one of the famous dish to j&k

Food Type: Veg Egg Non-Veg

Miscellaneous: Less spicy High Spicy

Close Update

Fig 6.2.20 Filter by Search

Momos Special
Phase 5, Mohali, Punjab

Dashboard Orders Menus Tax QR Code

Tax Management

SGST Enter SGST... %

CGST Enter CGST... %

SUBMIT

Momos Special Restaurant Admin © 2024 All rights reserved Made with ❤ by Sensation Solutions

Fig 6.2.21Add/Edit Taxes

Momos Special
Phase 5, Mohali, Punjab

QR Generation

+ Generate QR

Table - 01 Table - 02 Table - 03 Table - 04 Table - 05
Table - 06 Table - 07 Table - 08 Table - 09 Table - 10
Table - 12 Table - 13 Table - 14 Table - 15 Table - 16

Print Download Print Download Print Download Print Download
Print Download Print Download Print Download Print Download
Print Download Print Download Print Download Print Download

Momos Special Restaurant Admin © 2024 All rights reserved Made with ❤ by Sensation Solutions

Fig 6.2.22 QR Codes

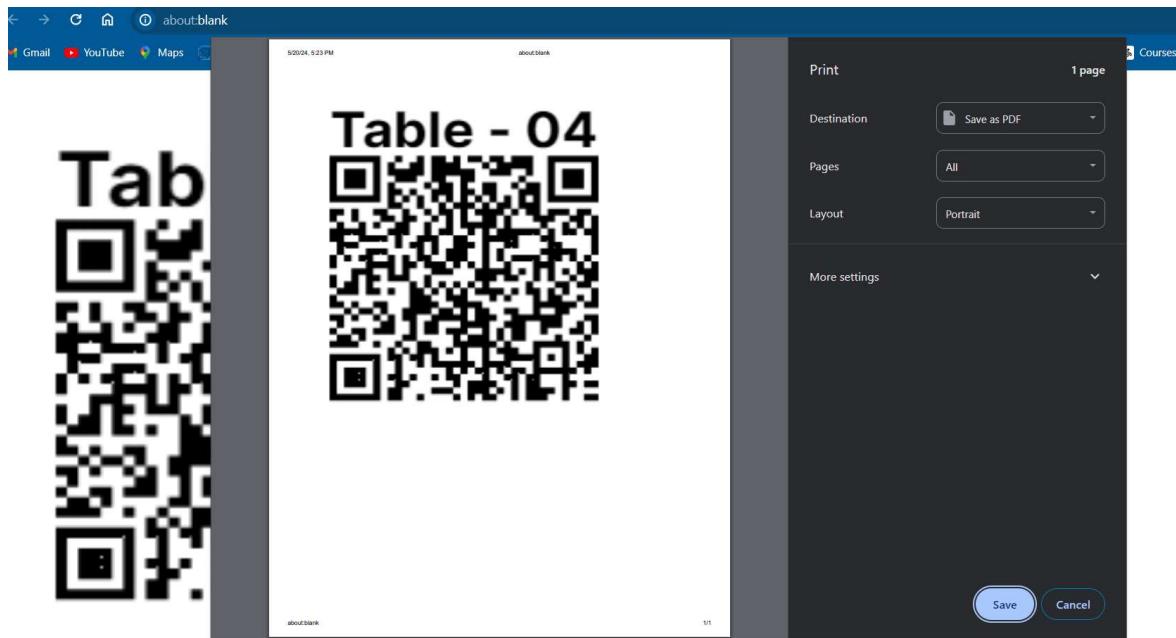


Fig 6.2.23 Print QR Code

Backend

Folder Structure or Repository of backend

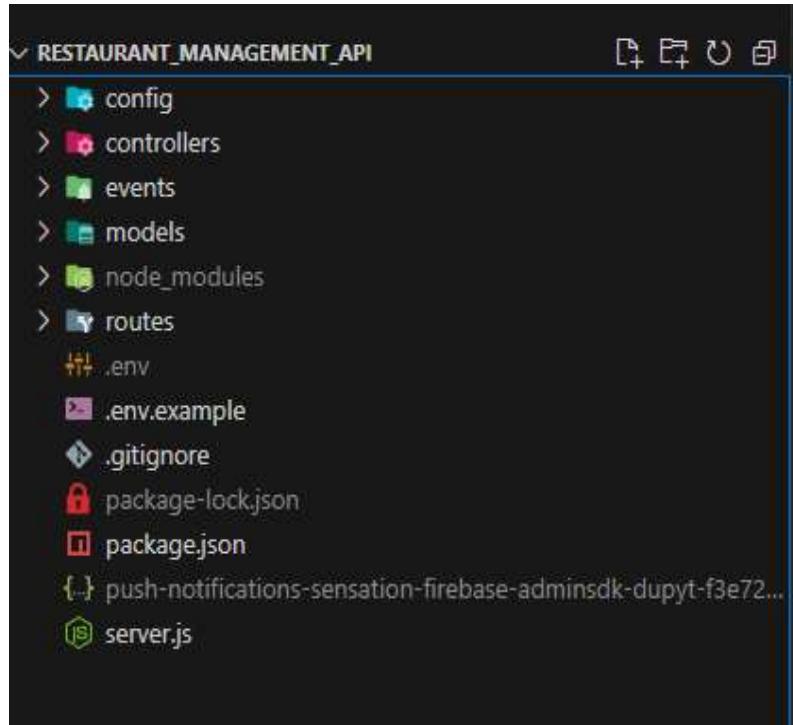


Fig 6.2.24 Folder Structure of Backend Repository

config:

- **Purpose:** The config folder is used to store configuration files that the application needs to function correctly.
- **Common Files:**
 - **Database Configuration:** Files like database.js might contain settings for connecting to databases, specifying connection strings, and defining other database-related settings.
 - **API Keys and Secrets:** Files that store sensitive information like API keys, secrets, and tokens. These might be split into different environments, e.g., config.dev.js for development, config.prod.js for production.

controllers:

- **Purpose:** Controllers manage the flow of data in the application and handle user interactions.
- **Common Functions:**

- CRUD Operations: Functions to Create, Read, Update, and Delete data, such as creating a new restaurant, retrieving a list of restaurants, updating restaurant information, and deleting a restaurant.
- Business Logic: Controllers also handle the business logic that processes the data before sending it to the client or the database.

models

- **Purpose:** Models define the structure of the data and interact with the database.
- **Common Components:**
 - **Schema Definitions:** Using libraries like Mongoose (for MongoDB), models define the structure of the data (schemas). For example, a Restaurant model might have fields like name, address, cuisine, and menu.
 - **Validation:** Models often include validation rules to ensure data integrity, such as requiring a restaurant name to be unique or an address to follow a specific format.
 - **Database Interactions:** Functions to interact with the database, such as saving new records, finding records, updating records, and deleting records.

node_modules

- **Purpose:** Contains all the installed Node.js dependencies (libraries and modules) required for the application to run.
- **Management:** Managed by the npm (Node Package Manager). When you run npm install, this folder gets populated with the necessary packages listed in package.json.

routes

- **Purpose:** Defines the endpoints (URLs) of the application and maps them to controller functions.
- **Common Structure:**
 - **Routing Files:** Each routing file handles a specific part of the application. For example, restaurantRoutes.js might define routes like /restaurants, /restaurants/:id, etc.
 - **Middleware Integration:** Routes can use middleware functions for tasks like authentication, logging, or validation before passing the request to the controller.

.env.example

- **Purpose:** Serves as a template for environment variable files.
- **Usage:** Developers copy this file to create a .env file and fill in the actual values. Environment variables might include DB_HOST, DB_USER, API_KEY, etc.
- **Environment Variables:** Used to store configuration settings that can change based on the environment (development, testing, production).

.gitignore

- **Purpose:** Tells Git which files and directories to ignore in version control.
- **Common Entries:**
 - **Node Modules:** node_modules/ to avoid committing large and frequently changing dependency files.
 - **Environment Files:** .env to prevent sensitive data from being exposed in the repository.
 - Logs and Build Artifacts: logs/, dist/, build/, etc.

package-lock.json

- **Purpose:** Locks the exact versions of each dependency installed.
- **Function:** Ensures that every developer working on the project and every deployment of the application uses the same versions of the dependencies, which helps avoid version conflicts.

package.json

- **Purpose:** Contains metadata about the project and its dependencies.
- **Key Fields:**
 - **name:** The name of the project.
 - **version:** The current version of the project.
 - **dependencies:** Lists the libraries the project depends on.
 - **scripts:** Defines script commands that can be run, like start, test, build, etc.

server.js

- **Purpose:** Acts as the main entry point for the application.
- **Common Tasks:**
 - **Setting Up the Server:** Includes code to initialize the server (often using frameworks like Express).
 - **Middleware Configuration:** Configures middleware for handling requests, parsing JSON, logging, etc.
 - **Routing:** Integrates routes so the server knows how to handle different endpoints.
 - Database Connection: Establishes connection to the database.

CHAPTER - 7

RESULTS AND DISCUSSIONS

Results

7.1 Enhanced Customer Experience: The implementation of the Restaurant Order Management Project has significantly improved the customer dining experience. The web-based platform provides a user-friendly interface that allows customers to browse menus, customize their orders, and leave reviews effortlessly. Key results include:

- **Increased Order Accuracy:** The digital system has reduced errors associated with manual order taking.
- **Faster Service:** Customers experience reduced wait times due to streamlined order processing.
- **Personalized Recommendations:** The system's ability to track customer preferences has led to more personalized dining experiences.

7.2 Operational Efficiency for Restaurants: using the platform have reported notable improvements in operational efficiency. The management tools provided by the system allow for easy menu updates, efficient order processing, and detailed performance analytics. Key outcomes include:

- **Time Savings:** Automation of order and billing processes has freed up staff time for other critical tasks.
- **Better Inventory Management:** Real-time tracking of orders and sales data has improved inventory management and reduced wastage.
- **Revenue Growth:** Enhanced operational efficiency and the ability to run targeted promotional campaigns have contributed to increased revenue.

7.3 Comprehensive Platform Administration Super admins: benefit from robust platform management tools that facilitate smooth operation and compliance. Key functionalities such as user account management, content management, and performance monitoring have led to:

- **Improved Oversight:** Comprehensive dashboards provide super admins with real-time insights into platform performance and user activity.
- **Enhanced Security:** Advanced security measures ensure that the platform remains secure and compliant with industry regulations.
- **Scalability:** The system's modular design allows for easy scalability to accommodate growing user bases and additional features.

Discussion

3.1 Addressing Inefficiencies in Traditional Restaurant Operations: The transition from traditional, manual processes to a digital, automated system has addressed several inefficiencies in restaurant operations. The project has demonstrated that integrating technology into restaurant management can lead to significant improvements in both customer satisfaction and operational efficiency. By reducing human error and streamlining workflows, restaurants can focus more on enhancing the overall dining experience.

3.2 The Role of Modern Technologies :The use of modern technologies such as HTML, CSS, JavaScript, React, MongoDB, Express, and Node.js has been pivotal in developing a robust and scalable platform. These technologies have enabled the creation of a responsive and intuitive user interface, efficient data handling, and real-time processing capabilities. The choice of these technologies aligns with the project's goals of providing a seamless and efficient system for all stakeholders.

3.3 Impact on Customer and Restaurant Dynamics: The platform has fundamentally changed the dynamics between customers and restaurants. Customers now have greater control and convenience when placing orders, while restaurants can more effectively manage their operations and engage with their clientele. This shift has the potential to foster stronger customer loyalty and drive repeat business.

3.4 Challenges and Future Directions: Despite the successes, there are challenges that need to be addressed. Ensuring consistent user engagement, managing system scalability, and maintaining security are ongoing concerns. Future enhancements could include the integration of advanced analytics for deeper insights into customer behavior, the incorporation of AI-driven recommendations, and the expansion of features to include delivery and takeout options.

3.5 Broader Implications for the Food Service Industry: The Restaurant Order Management Project serves as a model for how digital transformation can be applied within the food service industry. Its success underscores the importance of embracing technology to stay competitive in an increasingly digital world. The project's outcomes suggest that similar initiatives could be beneficial across other segments of the hospitality industry, promoting overall efficiency and customer satisfaction.

CHAPTER – 8

CONCLUSION AND FUTURE SCOPE

Conclusion

The **Digital Menu App Project** developed by Sensation Software Solutions Pvt Ltd has successfully addressed several critical inefficiencies in traditional restaurant operations by leveraging a modern, web-based platform. Key achievements of the project include:

- **Enhanced Customer Experience:** The platform provides a user-friendly interface that simplifies the ordering process, resulting in improved customer satisfaction and loyalty.
- **Operational Efficiency:** Restaurants have benefited from streamlined processes, better inventory management, and increased revenue, thanks to the efficient management tools integrated into the system.
- **Comprehensive Administration:** Super admins now have powerful tools for platform management, ensuring smooth operations, robust security, and regulatory compliance.

Overall, the project has demonstrated the transformative potential of technology in the food service industry, offering significant benefits to customers, restaurant operators, and platform administrators. By addressing the needs of all stakeholders, the project has laid a strong foundation for further innovations and improvements in the dining experience.

Future Scope

The success of the Restaurant Order Management Project opens several avenues for future development and enhancement. Key areas for future scope include:

1. Integration of Advanced Analytics and AI:

- Customer Insights: Implementing advanced analytics and machine learning algorithms to gain deeper insights into customer preferences and behaviors.
- Personalized Recommendations: Utilizing AI to offer more tailored dining recommendations and promotions based on individual customer profiles and past behavior.

2. Expansion of Service Offerings:

- Delivery and Takeout Options: Enhancing the platform to include comprehensive features for managing delivery and takeout orders, including real-time tracking and delivery optimization.
- Loyalty Programs: Developing integrated loyalty programs to reward frequent customers and encourage repeat business.

3. Mobile Application Development:

- Cross-Platform Availability: Creating mobile applications for iOS and Android to provide a seamless experience across all devices, enhancing accessibility and convenience for users on the go.

4. Enhanced Security Measures:

- Advanced Authentication: Implementing advanced authentication methods such as biometric verification and two-factor authentication to further secure user accounts and transaction data.
- Data Encryption: Strengthening data encryption protocols to protect sensitive information and ensure privacy.

5. Scalability and Performance Optimization:

- Infrastructure Upgrades: Investing in scalable cloud infrastructure to handle increased user loads and ensure smooth performance during peak times.
- Performance Monitoring: Continuous monitoring and optimization of the platform's performance to maintain high-speed, reliable service.

6. Global Expansion:

- Multilingual Support: Adding support for multiple languages to cater to a diverse, global customer base.
- Regional Customization: Customizing the platform to accommodate regional preferences and regulatory requirements, facilitating international expansion.

7. Enhanced User Engagement:

- Interactive Features: Introducing interactive features such as live chat support, social media integration, and community forums to enhance user engagement and satisfaction.
- Feedback Mechanisms: Developing robust feedback mechanisms to continuously gather user input and drive iterative improvements to the platform.

BIBIOGRAPHY

1. MongoDB, Inc. (2022). "Why Use MongoDB?" MongoDB Documentation. Retrieved from <https://www.mongodb.com/docs/>.
2. React Team. (2023). "Introducing React." React Documentation. Retrieved from <https://legacy.reactjs.org/docs/getting-started.html>.
3. Express. (2023). "Express - Node.js web application framework." Express Documentation. Retrieved from <https://expressjs.com/en/5x/api.html>.
4. Node.js Foundation. (2023). "About Node.js." Node.js Documentation. Retrieved from <https://nodejs.org/docs/latest/api/>.
5. W3C. (2023). "HTML & CSS Standards." World Wide Web Consortium (W3C). Retrieved from <https://www.w3.org/developers/>.
6. Google LLC. (2021). "Web Vitals: Essential metrics for a healthy site." Web.dev. Retrieved from [Web.dev](#).
7. <https://www.framer.com/motion/>